

### Remarks/Arguments

In the non-final Office Action dated August 6, 2008, it is noted that claims 1-12 are pending and that claims 1-12 stand rejected under either 35 U.S.C. §102 or 35 U.S.C. §103.

By this response, claim 5 has been amended to correct a grammatical error therein, and claims 5 and 6 have been amended to show proper antecedents and to clarify an aspect of the subject matter defined therein. Claims 13 and 14 have been newly added. The amendments are believed to be proper and justified. No new matter has been added.

In view of at least the following remarks, it is submitted that the claims pending in the application are novel and nonobvious. It is believed that this application is in condition for allowance. Reconsideration of the present application is respectfully requested.

#### *Amendment to the Claims*

Claims 5 and 6 have been amended to call for "a single storage medium." This recitation now shows the proper antecedent relationship with the respective independent base claim. In support of this amendment, only one storage medium is referenced in the specification. The sound, which is referenced by the sound identifier in the menu page composition segment in Table 1, is said to be stored in that storage medium as evidenced by the presence of the definite article on page 14, lines 1-6 of the specification. A similar disclosure is made on page 2 at lines 10-21.

The amendments to the claims are believed to be proper and justified. All the pending claims are believed to be supported by the original application. No new matter has been added to the claims.

#### *New Claims 13 and 14*

New claims 13 and 14 include additional distinguishing features over Piroumain which are fully supported by applicant's specification, for example table 1 where the "Exemplary menu page composition segment" gives an example of pseudo-code that defines a menu page. There are a number of buttons on the menu page, as defined by the parameter *num\_of\_buttons*. A loop with the loop parameter *button\_id* runs over all the buttons on the

page (as defined by the line "for *button\_id*=0; *button\_id* < *num\_of\_buttons*; *button\_id*++ {" ), and the loop ends in line 51 of the table (line 18 on page 8), defining every single button. That is, for decoding the bit-stream from the disc, the loop must be executed sequentially several times, separately for each of the buttons. Similarly, the remaining lines 19-24 of the table on page 8 define a number of graphical elements *num\_of\_graphics* (for example, "second menu items"), which are non-selectable and visible display data, as claimed.

### ***Cited Art***

The following references have been cited and applied as prior art in the present Office Action: "*Java™ GUI Development, The Authoritative Solution*", by Vartan Piroumian, pages 12, 19, 225, 227-229, and 232 (Sams 1999) (hereinafter referenced as "*Piroumian*") and "*jiGUI-Java Music Player*", comprising one (1) web page at the cited address bearing a date of April 1, 2002 (hereinafter referenced as "*jiGUP*").

### ***Rejection of Claims 1-4 and 7-11 under 35 U.S.C. §102***

Claims 1-4 and 7-11 stand rejected under 35 U.S.C. §102 as being anticipated by Piroumian. This rejection is respectfully traversed.

In the present application, claim 1 is an independent method claim that serves as a base claim for claims 3-4 and 7-8. Claim 2 is an independent apparatus claim that includes limitations substantially similar to claim 1 and serves as an independent base claim for claims 9-11. For the sake of brevity and in view of the claim limitation similarities, the discussion below will focus on claim 1.

The present invention provides a new type of data object, and a corresponding flexible decoding method. This new type of data object allows combinations of visible, invisible, selectable, and non-selectable menu objects. As a particular advantage of the claimed subject matter, a menu including at least visible selectable and visible non-selectable objects, or even all the different types of menu objects, can be generated from a single data structure. That is, the menu objects, including visible, invisible, selectable and non-selectable objects, use the same data structure and decoder, which is not possible with the known prior art.

Contrary to the assertions made on pages 8 and 9 of the present Office Action in the *Response to Arguments* in reliance upon additional information about Java cited but not applied in the Office Action, Piroumian even as supplemented by the additional cited information does not show “selectable and invisible menu items,” as defined in the claims. The *No-arg constructor* identified with the *JMenuItem()* construct in Piroumian on page 232 appears to generate “a menu item with no defined text or icon.” The item constructed by the *No-arg constructor* appears to be a menu item having no defined text or icon associated with it. This menu item appears to be empty because it has neither icon nor text. But it is understood that the menu item created by the *No-arg constructor* is visible, contrary to the limitations in the claims. There is no teaching or suggestion that the menu item so created is not visible.

In the *Response to Arguments*, particularly in paragraph 8 of the present Office Action, it is mentioned that,

“[a] reference to the *no-arg constructor* on page 232 is to indicate that a *JMenu* item can be constructed without further adding it to a frame such that those components would be packed (pg. 228) and then *setVisible* (pg. 228). A component, such as a *JMenu* item that is not added to be *setVisible* would still nonetheless be constructed but would only have logical and not visible representation.”

There is nothing, either logically or specifically from the documentary evidence, to support the latter portion of the USPTO’s assertion reproduced above. In the remarks below, it will be shown that a *JMenuItem* has always a visible representation. Further, with respect to the *Response to Arguments* concerning the alternative involving Figures 7.14 and 7.15 of Piroumian, the invisible menu items shown in those figures are not selectable, while they are invisible. The invisible menu items therein must first be made visible before they can be selected.

Using the applied references and the supplementary references provided by the USPTO, an example of a menu resulting from the use of such a *No-arg constructor* has been programmed in Java Swing, which is described at least on the Java web site having the address <http://java.sun.com/docs/books/tutorial/uiswing/components/menu.html>, for example. The Java Swing code program and the resulting menu are shown below. The programmed menu in Figures 1 and 2 includes: a first button identified with text as “*item1*,” a second button created with the *No-arg constructor*, and a third button identified with text as “*item3*.”

The second button, which was created with the *No-arg constructor* “*JMenuItem()*”, appears empty having neither an icon nor text. It should be noted that the second button is visible between “*item1*” (the first button) and “*item3*” (the third button). For better recognition of the second button, that button, the empty button in the image, was selected as shown in Figure b. In Figure a, the first button identified as “*item1*” is shown to be selected.

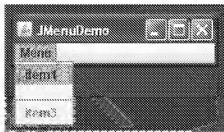


Figure a

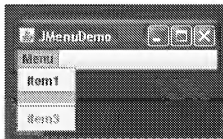


Figure b

The Java Swing code that generated the three separate menu items is given as follows:

```
JMenuItem item1 = new JMenuItem("item1");  
JMenuItem item2 = new JMenuItem();  
JMenuItem item3 = new JMenuItem("item3");
```

The code relates to both Figures a and b shown above. The full code listing for the example shown in Figures 1 and 2 is shown immediately below with the code listing above shown in bold below:

```
import java.awt.Dimension;  
import javax.swing.*;  
  
public class JMenuDemo extends JFrame {  
  
    public JMenuDemo() {  
        super("JMenuDemo");  
    }  
  
    public static void main(String[] args) {  
        JFrame frame = new JMenuDemo();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        JMenuItem item1 = new JMenuItem("item1");  
        JMenuItem item2 = new JMenuItem();  
        JMenuItem item3 = new JMenuItem("item3");  
  
        JMenu menu = new JMenu("Menu");  
        menu.add(item1);  
        menu.add(item2);
```

```
        menu.addSeparator();  
        menu.add(item3);  
        item3.setEnabled(false);  
  
        JMenuBar bar = new JMenuBar();  
        bar.setPreferredSize(new Dimension(200, 20));  
        bar.add(menu);  
        frame.setJMenuBar(bar);  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

From this example, it is clear that, contrary to the USPTO's assertion, the second menu item created from the No-arg constructor *JMenuItem* always has a visible representation. Thus, it is clear from this example that Piroumian does not teach, show, or suggest this limitation in the independent claims.

Concerning the claimed limitation "wherein the third menu items are menu buttons that are automatically activated upon selection," the present Office Action states that this limitation is met by the "*void setAccelerator (Keystroke keystroke)*" element described by Piroumian. However, as indicated by the name of the element, the element is usable for accelerating the access to a button via one or more keyboard strokes as opposed to selection with a cursor. There is nothing in the description of this method/constructor to indicate that it pertains to invisible buttons. It is believed that the button (menu item) to which this method/constructor is applied always has associated graphic representation data so that it can be selected graphically. This means that Piroumian teaches a method/constructor that is not suggestive of claimed third menu items. That is, the Piroumian does not show that "the third menu items are menu buttons that are automatically activated upon selection", wherein the third menu items are invisible, as defined in the claims.

In view of the remarks above, it is believed that Piroumian does not teach, show, or suggest all the limitations in independent base claims 1 and 2. It is submitted that Piroumian neither anticipates nor makes obvious independent claims 1 and 2 and the respective claims dependent thereon. Hence, it is believed that claims 1 and 2 and dependent claims 3-4 and 7-11 are allowable under 35 U.S.C. §102 and 35 U.S.C. §103. Withdrawal of this rejection is respectfully requested.

In addition to the reasons set forth above with respect to claim 1, it is noted that, with respect to claims 8 and 11, Piroumian fails to expressly teach that “the first and the second menu items have associated display positions comprising a horizontal address and a vertical address.” Therefore, it is submitted that Piroumian neither anticipates nor makes obvious claims 8 and 11. Hence, for these additional reasons, it is believed that claims 8 and 11 are allowable under 35 U.S.C. §102 and 35 U.S.C. §103. Withdrawal of this rejection is respectfully requested.

***Rejection of Claims 5, 6, and 12 under 35 U.S.C. §103***

Claims 5, 6, and 12 stand rejected under 35 U.S.C. §103 as being unpatentable over Piroumian in view of jIGUI. This rejection is respectfully traversed.

In the present application, claim 1 is an independent method claim that serves as a base claim for claims 5 and 6. Claim 2 is an independent apparatus claim that includes limitations substantially similar to claim 1 and serves as an independent base claim for claim 12.

The jIGUI reference appears to be a brief advertisement for a new release of the software, namely, version 2.1.1. It appears to document certain features of a Java music player. The present Office Action, at page 6, states that “jIGUI discloses a Java Applet, wherein sound data are associated to a state of a menu button, the sound data and the menu data segment being read from the same storage medium and being played back upon entry of the button into the associated state (pg. 1.)” But, as stated in the prior response, support for this assertion in its entirety does not appear anywhere on this webpage.

Even if the *Response to Arguments* on page 9 of the present Office Action were to be construed in a light most favorable to the USPTO's position, they still fail to identify where in the jIGUI reference there exists any teaching, showing, or suggestion for “the sound data and the menu data segment being read from a single storage medium,” as defined in claim 5, or for “audio-visual data stored on a single storage medium with the menu data segment”, as defined in claim 6, or for “the sound data ... being read from the same storage medium as the menu data segment”, as defined in claim 12. Although the jIGUI reference is only 1 page long, it would appear to be necessary for the Examiner to identify with specificity where any one of

the above identified limitations is met by jIGUI. In *Response to Arguments* on page 9 of the present Office Action, the screenshot from the jIGUI reference was used to support a portion of the rejection dealing with the actuation of certain buttons on the Java Music Player. But neither the screenshot nor any other part of jIGUI discuss where sound data or audio-visual data or menu data segment are stored. It would appear to be quite reasonable to expect that sound or audio-visual data are stored in a place different from the menu data segment when dealing with a music player. For example, sound or audio-visual data for the Java Music Player could be stored on a CD, DVD, some type of flash memory device, or even a remote server for a streaming broadcast shown in the screenshot, all being storage remote from and different from the storage medium for the menu data segment defined in the claims. Thus, the sound data, audio-visual data, and the menu data segment are not taught, shown, or suggested to be stored in "a single storage medium" defined in the claims

In addition to the deficiencies noted above with respect to the limitations in claims 5, 6, and 12, it should be understood that the jIGUI reference fails to cure the deficiencies already noted above with respect to the independent base claims, namely claims 1 and 2. As a result, neither Piroumian nor jIGUI, separately or in combination, appear to teach, show, or suggest all the elements in dependent claims 5, 6, and 12.

In view of the remarks above and the remarks concerning claims 1 and 2 in the prior section of this response, it is submitted that dependent claims 5, 6, and 12 would not have been obvious to a person skilled in the art upon a reading of Piroumian and jIGUI, either separately or in combination. Hence, it is believed that dependent claims 5, 6, and 12 are allowable under 35 U.S.C. §103. Withdrawal of this rejection is respectfully requested.

### ***Conclusion***

In view of the foregoing, it is respectfully submitted that all the claims pending in this patent application are in condition for allowance. Reconsideration and allowance of all the claims are respectfully solicited.

If, however, the Examiner believes that there are any unresolved issues requiring adverse final action in any of the claims now pending in the application, it is requested that the Examiner contact the applicant's attorney at (609) 734-6813, so that a mutually convenient

date and time for a telephonic interview may be scheduled for resolving such issues as expeditiously as possible.

In the event there are any errors with respect to the fees for this response or any other papers related to this response, the Director is hereby authorized to charge any shortages and credit any overcharges of any fees required for this submission to Deposit Account No. 07-0832.

Respectfully submitted,

By: /Reitseng Lin/  
Reitseng Lin  
Reg. No. 42,804  
Phone (609) 734-6813

Patent Operations  
Thomson Licensing Inc.  
P.O. Box 5312  
Princeton, New Jersey 08540  
November 5, 2008